Name: _____

Student ID: _____

Department of Computer Science

Final Exam, CS 411a/538a — Databases II

Prof. S. Osborn
Dec. 16, 2003
3 Hours
One sheet of notes allowed
No electronic aids

Answer all questions on the exam page

This paper contains 16 pages; the last page is for rough work.

Question	Maximum	Your Mark
1	18	
2	28	
3	24	
4	29	
5	18	
6	16	
7	22	
8	12	
Total	167	

1. (18 marks) For each of the following, state whether the term applies to one, two or all three of: **relational databases, object-oriented databases or XML databases**. Give a **BRIEF** reason for your answer.

   (a) attribute

   (b) sharing

   (c) closed query language

   (d) concurrency control

   (e) nesting of data to arbitrary levels

   (f) always has a schema

   (g) tags in the data

   (h) identity comparisons

   (i) path expressions

2. (28 marks) Each of the following statements, state whether it is **true** or **false**. If it is false, correct the statement without changing the underlined text.

   (a) In an OODB, bytes that are stored together would include nested subobjects. **T** **F**

   (b) Vertical Fragmentation is undertaken in an distributed database in order to have data local to where it is frequently used. **T** **F**

   (c) The statistic distinct values of A records the number of tuples in a relation. **T** **F**

   (d) Pointer swizzling is necessary because compiler writers don't know anything about databases. **T** **F**

   (e) As far as ACID Properties are concerned, A is the responsibility of concurrency control. **T** **F**

   (f) As far as ACID Properties are concerned, C is the responsibility of concurrency control. **T** **F**

   (g) Serializable execution of a set of concurrent transactions guarantees that transactions never abort. **T** **F**

   (h) Strict two-phase locking means that deadlocks never occur. **T** **F**

(i) Wound-Wait is a method of using timestamps as the recovery method for distributed databases. **T** **F**

(j) Write-ahead logging is a concurrency control technique which may require redo, but never requires undo. **T** **F**

(k) Blocking occurs in two-phase commit when a participant has voted yes, and times out waiting to hear back from the coordinator. **T** **F**

(l) The GRANT and REVOKE statements in DB2 are examples of Mandatory access control. **T** **F**

(m) The no-read-up principle in Mandatory Access Control means that a user can read at exactly their own level, and no other. **T** **F**

(n) Polymorphism is sometimes necessary to keep secrets. **T** **F**

3. (24 marks) Consider the following XML document:

```
<?xml version="1.0"?>
<ContestantSet>
        <Contestant Name = "Shania Twain" Hometown = "Timmins">
                <Repertoire>
                        <SongTitle>Forever and For Always</SongTitle>
                        <SongTitle>Thank You Baby</SongTitle>
                        <SongTitle>Come on Over</SongTitle>
                </Repertoire>
                <Performance>
                        <SongTitle>Come on Over</SongTitle>
                        <SongTitle>Forever and For Always</SongTitle>
                        <SongTitle>Thank You Baby</SongTitle>
                </Performance>
        </Contestant>
        <Contestant Name="Paul McCartney" Hometown = "Liverpool">
                <Repertoire>
                        <SongTitle>Bah Bah Black Sheep</SongTitle>
                        <SongTitle>Band on the Run</SongTitle>
                        <SongTitle>Michelle</SongTitle>
                        <SongTitle>Yesterday</SongTitle>
                </Repertoire>
                <Performance>
                        <SongTitle>Bah Bah Black Sheep</SongTitle>
                        <SongTitle>Michelle</SongTitle>
                        <SongTitle>Tomorrow</SongTitle>
                </Performance>
        </Contestant>
</ContestantSet>
```

Assume the following text appears in a context file:

```
define global $contestants { treat as document ContestantSet
(glx:document-validate("contestants.xml", "ContestantSet")) }
```

(a) (2 marks) Which of the following is an example of an attribute?
i Name = "Shania Twain"
ii <SongTitle>Forever and For Always</SongTitle>
iii </Contestant>
iv none of the above

(b) (2 marks) Which of the following is an example of a subelement?
i Name = "Shania Twain"
ii <SongTitle>Forever and For Always</SongTitle>
iii </Contestant>
iv none of the above

(c) (3 marks) List all of the tags used in the above document.

(d) (3 marks) Give an **XPath** expression to list song titles which appear in a Performance.

(e) (4 marks) Give an **XPath** expression to list contestant names where the contestant has a song in their repertoire which contains "Ba".

(f) (4 marks) Give an **XQuery** expression to list all contestants whose name contains the letter "a". The answer should give contestant names and hometowns.

(g) (6 marks) Give an **XQuery** expression which generates a valid XML document, containing a contestant set, where each contestant has a name, and a list of song titles such that the song title is in both the Repertoire and the Performance for this contestant in the original document.

4. (29 marks) Consider the following relations for a relational database:

```
Singer(Name, Hometown, DofB), key is (Name, DofB)
Song(Title, Composer, YearComposed), key is (Title, Composer)
Repertoire(Name, Title, YearLearned), key is (Name, Title)
Performance(Name, Title, Position), key is (Name, Title)
```

   (a) (5 marks) We wish to compute the join of Singer with Repertoire, but the relations are on different sites of a network in a distributed database. Explain the steps of the semi-join strategy which computes this join and leaves the result at the site of Singer.

   (b) Now, consider the following query:

$$\pi_{Name,Title}(\sigma_{Title\ contains"Ba"\wedge Singer.Name=Repertiore.Name}(\sigma_{\wedge Repertoire.Name=Performance.Name\wedge Repertoire.Title=Performance.Title}(Singer\times(Repertoire\times Performance))))$$

      i. (3 marks) Express this query in English.

      ii. (10 marks) Show how the algebraic query optimization technique would optimize the following query (you may just show the original tree and the final tree if you like). **Do Not change the order of the $\times$'s**. Indicate in your final tree where $\bowtie$'s may be substituted for $\times$'s. There is more space for this answer on the next page.

(extra space for the answer to question 4, b, ii)

(c) Consider these relations again:

```
Singer(Name, Hometown, DofB), key is (Name, DofB)
Song(Title, Composer, YearComposed), key is (Title, Composer)
Repertoire(Name, Title, YearLearned), key is (Name, Title)
Performance(Name, Title, Position), key is (Name, Title)
```

i. (2 marks) If we have referential integrity between Repertoire and Singer (i.e. we know that every Name value in Repertoire is also in Singer), what can we say about the number of distinct values of Name in Repertiore?

ii. (4 marks) We want to compute the query:
Q = $\pi_{Name,Title}$(Repertoire) $\cup$ $\pi_{Name,Title}$(Performance)
Give upper and lower bounds for the number of distinct values of Name in Q, in terms of the number of number of distinct values of Name in the Repertoire relation and the number of distinct values of Name in the Performance relation.

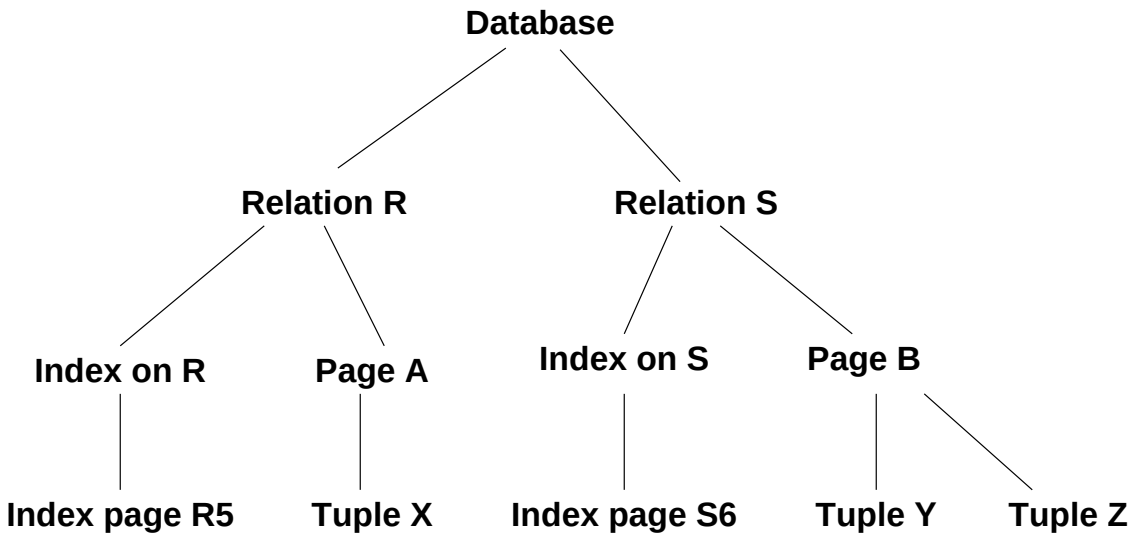
iii. (5 marks) This SQL query results in a "bag union":

```
Select Title from Repertoire
union
Select Title from Performance
```

Describe an algorithm for executing this "bag union".

5. (18 marks) Put your answer to this question on the next page.

   Consider the following (simplified) tree of granules for a relational database:



   (a) (6 marks) We have transaction T1: tuple Z is to be deleted from page B of relation S. The transaction consists of reading the index page S6 to find Z, and then doing the delete of tuple Z. Show a sequence of lock and unlock instructions which follow the rules for multiple granularity locking, and which allows this transaction to execute.

   (b) (6 marks) Transaction T2: this transaction wants to read tuple Y in Relation S, after first reading the index page S6. It then wants to update an attribute of Tuple X, after first reading index page R5 in relation R. Show a sequence of lock and unlock instructions which follow the rules for multiple granularity locking, and which allows this transaction to execute.

   (c) (6 marks) Show a valid interleaving of T1 and T2 such that there are no deadlocks. Use the following lock compatibility matrix. Put your answer on the next page.

Lock Compatibility Matrix

	Already Granted					
Requested	None	IS	IX	S	SIX	X
IS	yes	yes	yes	yes	yes	-
IX	yes	yes	yes	-	-	-
S	yes	yes	-	yes	-	-
SIX	yes	yes	-	-	-	-
X	yes	-	-	-	-	-

(space for answer to question 5.)

6. (16 marks) Suppose the following timestamps are recorded for the following data items:

data item	read TS	write TS
x	19	25
y	25	19
z	21	21
w	25	25

For each of the following operations, state whether or not it would be allowed using the revised timestamp ordering algorithms for concurrency control, and, if allowed, **whether or not any timestamps change**. Assume all these operations are independent of each other (i.e., they all refer to the original timestamps for x, y, z and w). If any of them uses the Thomas Write Rule, say that.

(a) read z on behalf of transaction T whose timestamp is 22.

(b) write z on behalf of transaction T whose timestamp is 22.

(c) read y on behalf of transaction T whose timestamp is 22.

(d) write y on behalf of transaction T whose timestamp is 22.

(e) read x on behalf of transaction T whose timestamp is 22.

(f) write x on behalf of transaction T whose timestamp is 22.

(g) read w on behalf of transaction T whose timestamp is 22.

(h) write w on behalf of transaction T whose timestamp is 22.

7. (22 marks) Consider the diagram for Two-Phase commit reproduced here from the Özsu and Valduriez book:

   (a) (4 marks) Which nodes represent phase one of the Two-Phase commit? Give nodes for both the coordinator and the participants.

   (b) (4 marks) Which nodes represent phase two of the Two-Phase commit? Give nodes for both the coordinator and the participants.

(c) (2 marks) Is it possible for the coordinator to be in the COMMIT state, and a participant to be in the READY state at the same time?

(d) (2 marks) Is it possible for the coordinator to be in the COMMIT state, and a participant to be in the ABORT state at the same time?

(e) (3 marks) In what nodes can the coordinator time out?

(f) (3 marks) In what nodes can the participant time out?

(g) (4 marks) Does this diagram show what the participant's algorithm is if the coordinator crashes? If so, what state is that? If not, explain briefly what happens.

8. (12 marks) Consider the following multilevel relation in the **SeaView** model (assume security levels TS > S > C > U):

Name	C_{Name}	Department	C_{Dept}	Salary	C_{Salary}	TC
Bob	C	Dept1	C	10K	C	C
Bob	U	Dept3	U	20K	S	S
Ann	S	Dept2	S	30K	TS	TS
Sam	TS	Dept3	TS	30K	TS	TS

(a) (6 marks) Give the C-instance (all data visible to C-level users) of the above relation.

(b) (4 marks) Can a user cleared at level C insert a tuple for Bob with Dept1 and salary of 15K? If so, show all the attributes of this tuple (including the C_{Name}, C_{Dept} and C_{Salary} and TC. If not, explain why not.

(c) (2 marks) At what level(s) can the whole relation above be labeled?

(for rough work)